# Kernel pour la compétition Kaggle:

# Lyft Motion Prediction for autonomous Vehicles

Marco Berta

# Table des Matiéres

# 1. Introduction

Le projet a été réalisé dans la cadre de la compétition Kaggle sponsorisée par la société de Véhicule de Tourisme avec Chauffeur (VTC)  Lyft basée aux États Unis. (*Lyft Motion Prediction for Autonomous Vehicles | Kaggle*, no date)
L'entreprise de VTC a mis à disposition de la communauté Kaggle plus de 55 000 images 3D annotées, des données provenant de 7 caméras et 3 Lidar et une cartographie HD réalisée par ses véhicules.
La modélisation des données idéalement pourra aider à construire un véhicule totalement autonome, que n'aura pas besoin d'un chauffeur humain.

L'objectif du modèle est la prédiction des trajectoires des véhicules détectés, que permet de décider la meilleure trajectoire pour la destination choisie.  Lyft propose un modèle du départ basé sur une réseaux des neurones convolutionelle, ResNet50(*l5kit/agent_motion_prediction.ipynb at master · lyft/l5kit*, no date) et un logiciel propriétaire, pythonl5kit, que permet d'importer les données avec le langage de programmation Python.  Des paquets additionnels dans l5kit permettent d'estimer l'erreur de prédiction (*l5kit.evaluation package — L5Kit 1.1.0 documentation*, no date).
Les données sont optimisées pour le paquet PyTorch, et celui à été utilisé pour la modèlisation.

Dans ce projet un nouveau modèle est proposé et les paramétrés sont optimisés pour obtenir le plus faible erreur et la meilleure note possible sur la leaderboard de la compétition. Les contraintes principaux sont la puissance de calcul limité fournie par les machines virtuels de Kaggle et la compatibilité entre l5kit et PyTorch.

# 2. Exploration des données

Les données d'identification des objets sur la route sont collectées par des cameras, pour chaque objet la distance est mesurée par des lidars placées sur des voitures autonomes (Figure 1)

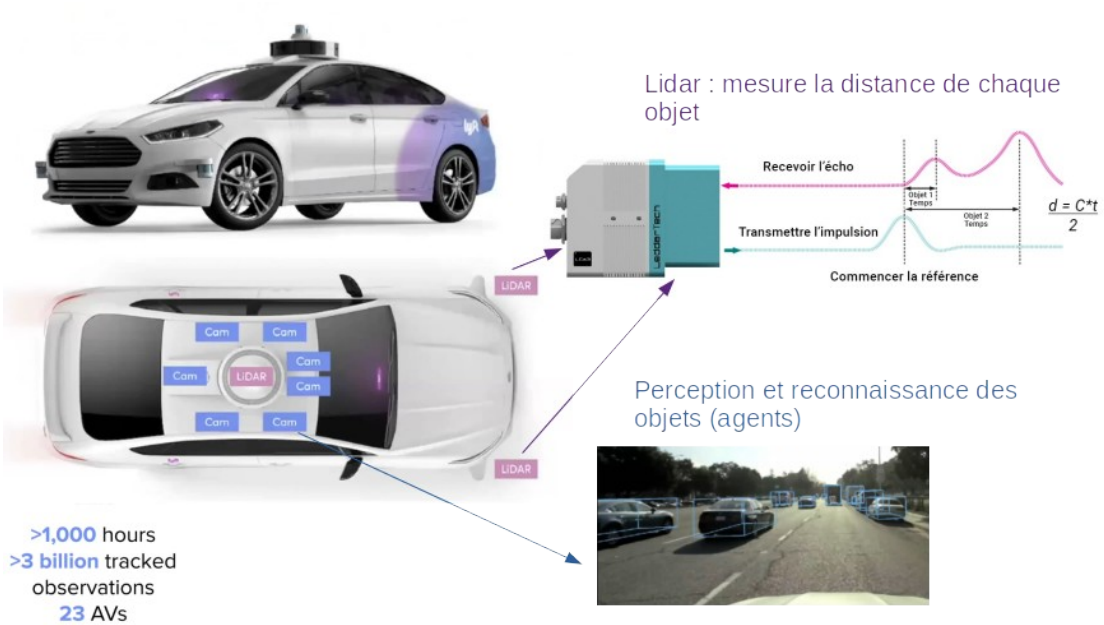


Fig. 1 Collection des données.

Les objets détectés sont intégrés dans des  images prises par un satellite pour obtenir des cartes sémantiques comme celle en Figure 2.

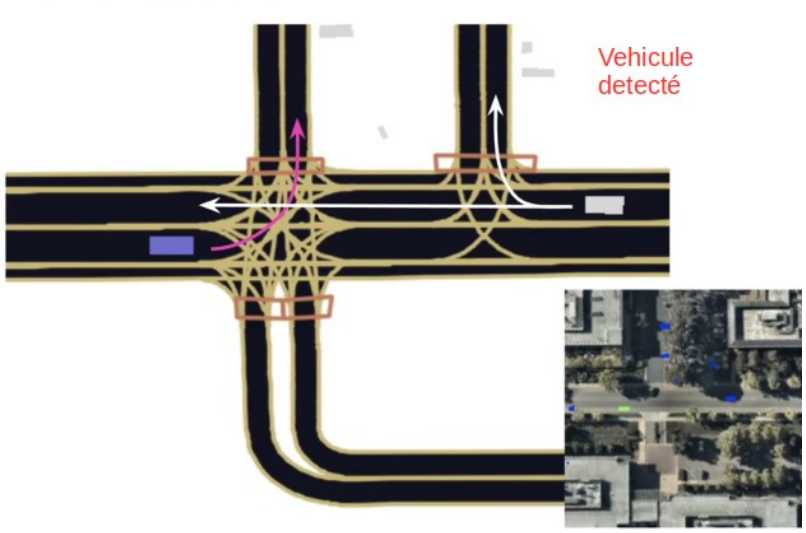


Fig. 2 Carte sémantique dérivée des images du satellite intégrées avec .

Un ensemble des cartes sémantiques correspond à une scène,  25 seconds de vidéo. Tous les scènes sont en suite stockées dans des fichiers en format zarr (similaire à un matrice numpy) que peuvent être traités par PyTorch.

| Num Scenes | Num Frames | Num Agents | Num TR lights | Total Time (hr) | Avg Frames per Scene | Avg Agents per Frame | Avg Scene Time (sec) | Avg Frame frequency |
|---|---|---|---|---|---|---|---|---|
| 100 | 24838 | 1893736 | 316008 | 0.69 | 248.38 | 76.24 | 24.83 | 10.00 |

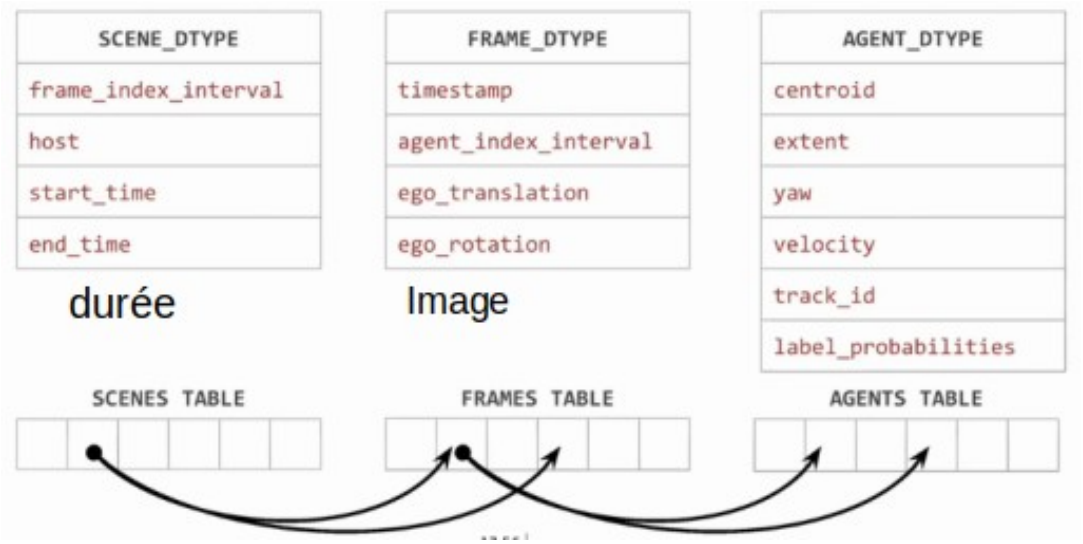


Fig. 3 Structure des données dans les fichiers ‘zarr’.

Dans chaque scène on trouve un liens vers un tableau d'images, et chaque image contient plusieurs objets/agents  (Fig.3).

Des fichiers zarr correspondent respectivement aux données d'entraînement,de validation et de test.

L' objectif du modèle entraîné est de prédire les 50 images (frames) successives a ceux modélisées.

# 3. Modélisation initiale et amélioration

Au début de la compétition Kaggle un exemple basilaire de modélisation était fourni par les sponsors(*l5kit/agent_motion_prediction.ipynb at master · lyft/l5kit*, no date). Les données étaient modélisées par une réseaux des neurones ResNet50(*[GPU Training] LyFt: Torch Baseline | Kaggle*, no date). Les auteurs n'ont pas donné une évaluation sauf la note finale dans la leaderboard : 1999. La note est inversement proportionnelle à la précision de la solution.

Pour obtenir un meilleure résultat plusieurs stratégies ont été adoptées. La plus simple, est la modification de l'architecture ResNet avec plus des couches entièrement connectée et/ou plus des couches de convolution.

Les graphiques en Figure 4 montrent que la fonction de perte ''non-negative likelyhood'' utilisée par le l5kit est similaire et une simple modification de la structure ResNet n'est pas suffisante pour obtenir une amélioration significative.

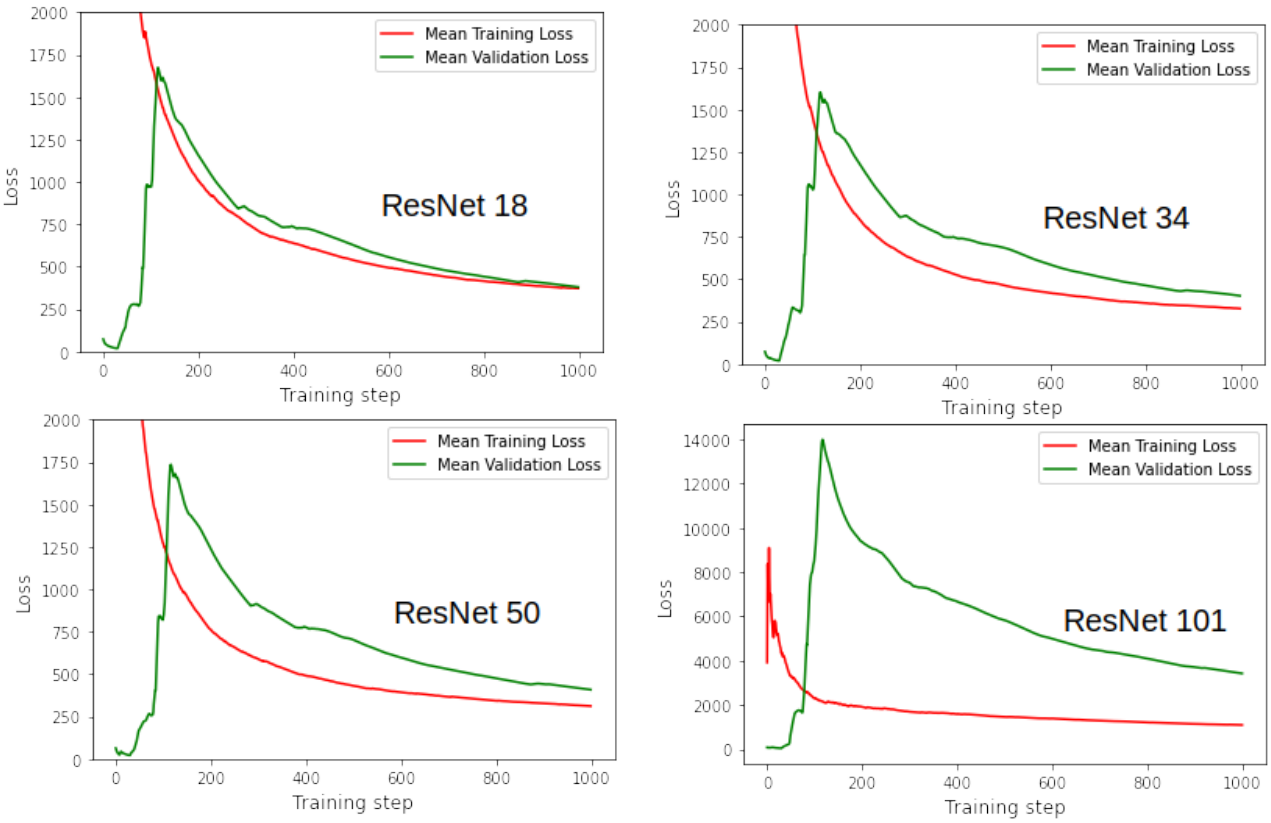


Fig. 4 Fonction de perte « non-negative-likelyhood » pour differentes structures ResNet

Un meilleur résultat à été obtenu avec la modélisation avec la fonction de perte « mean square error », une réduction de la taille des groupes (‘‘batches’’) d’images modélisées compensée par une adaptation de la vitesse d’apprentissage (learning rate)(Huynh, no date). La performance était meilleure avec le changement d’architecture de la réseaux des neurones, EfficientNet au lieux de ResNet (Figure 5). La note finale obtenue était 1172.

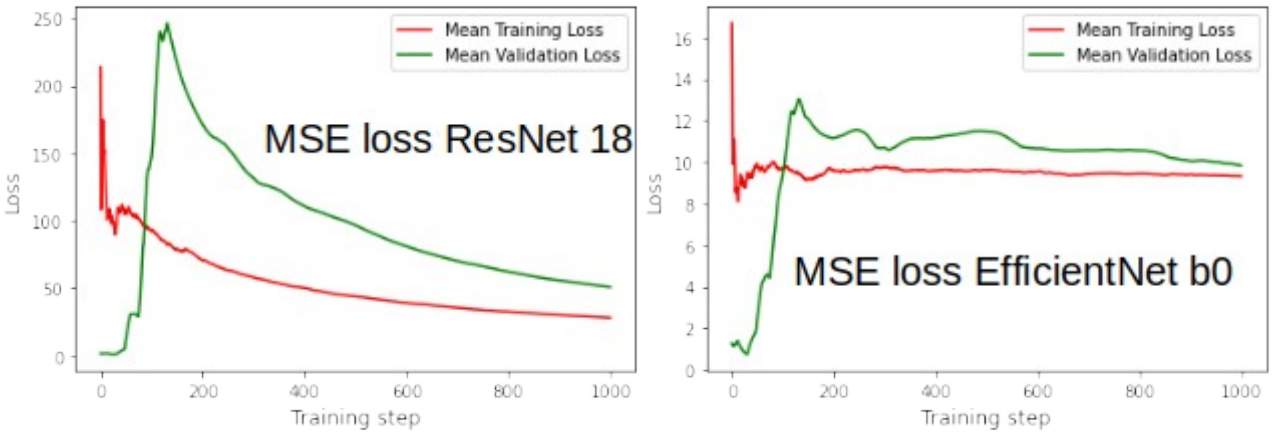


Fig 5. Resnet18 vs. Efficentnet-b0

# 4. Conclusion

Une amélioration de la note du kernel (1172 vs 1999) était obtenue avec le changement de réseaux des neurones, EfficientNet au lieux de ResNet et l’optimisation de paramétrés (pixel size, optimizer, batchsize, learning rate). L’utilisation de l’ accélération  avec TPU n’a pas donné un fichier à soumettre à cause de l’instabilité de la machine virtuelle utilisée sur Kaggle.

https://www.kaggle.com/mclikmb4/lyft-pytorch-efficientnet-b0-on-tpu

# 5. Bibliographie

*[GPU Training] LyFt: Torch Baseline | Kaggle* (no date). Available at: https://www.kaggle.com/rhtsingh/gpu-training-lyft-torch-baseline (Accessed: 7 October 2020).

Huynh, D. (no date) ‘How to get 4x speedup and better generalization using the right batch size’, *Medium*. Available at: https://towardsdatascience.com/implementing-a-batch-size-finder-in-fastai-how-to-get-a-4x-speedup-with-better-generalization-813d686f6bdf (Accessed: 7 October 2020).

*l5kit.evaluation package — L5Kit 1.1.0 documentation* (no date). Available at: https://lyft.github.io/l5kit/API/l5kit.evaluation.html (Accessed: 5 October 2020).

*l5kit/agent_motion_prediction.ipynb at master · lyft/l5kit* (no date). Available at: https://github.com/lyft/l5kit/blob/master/examples/agent_motion_prediction/agent_motion_prediction.ipynb (Accessed: 5 October 2020).

*Lyft Motion Prediction for Autonomous Vehicles | Kaggle* (no date). Available at: https://www.kaggle.com/c/lyft-motion-prediction-autonomous-vehicles (Accessed: 5 October 2020).